

Vector Fields for Robot Navigation Along Time-Varying Curves in n -dimensions

Vinícius M. Gonçalves, Luciano C. A. Pimenta, Carlos A. Maia, Bruno C. O. Dutra, and Guilherme A. S. Pereira

Abstract—This paper presents a methodology for computing artificial vector fields that allows a robot to converge to and circulate around generic curves specified in n -dimensional spaces. These vector fields may be directly applied to solve several robot navigation problems such as border monitoring, surveillance, target tracking and multi-robot pattern generation, with special application to fixed-wing aerial robots, which must keep a positive forward velocity and cannot converge to a single point. Unlike previous solutions found in the literature, the approach is based on fully continuous vector fields and is generalized to time-varying curves defined in n -dimensional spaces. We provide mathematical proofs and present simulation and experimental results that illustrate the applicability of the proposed approach. We also present a methodology for constructing the target curve based on a given set of its samples.

Index Terms—Mobile robots, motion control, limit cycles, time-varying systems.

I. INTRODUCTION

Vector field based approaches have been extensively used to guide and control mobile robots in the execution of different tasks. This is mainly due to an important characteristic of such methods that is the integration of path planning, trajectory planning, and robot control in the same approach, which allows for stability proofs, and real world implementations [1]. Given a n -dimensional domain, Ω , a vector field \mathbf{h} is defined such that $\mathbf{h} : \Omega \rightarrow T_{\mathbf{q}}(\Omega)$, where $T_{\mathbf{q}}(\Omega)$ is the tangent space of Ω , and $\mathbf{q} \in \Omega$. In robotics, the domain, Ω , is the robot's configuration space, also represented by \mathcal{C} . The desired task is then accomplished by forcing the robot to use the vector field as velocity or acceleration inputs.

A classical problem is to drive a single robot from an initial configuration, \mathbf{q}_0 , to a final configuration, \mathbf{q}_f . This problem was solved by means of several artificial vector fields such as the ones proposed in [2], [3], [1], [4], [5], [6], and [7].

Other important problems that were recently considered by several researchers are pattern generation and pattern tracking. Different tasks, such as surveillance, manipulation, and boundary monitoring can be executed by using solutions of these problems. A very interesting application is, for instance, shown in [8], where a flock of aerial vehicles is used to track a chemical plume released in the atmosphere.

The *pattern generation problem* involves controlling a team of mobile robots in order to converge to a pre-specified static geometric pattern. On the other hand, the *pattern tracking*

problem requires one or more vehicles that are controlled to converge to a closed pattern and to circulate along it. The two problems are essentially very similar and so are the solutions found in the literature.

In [9], [10] and [11], vector fields are computed to solve the pattern generation problem for static two-dimensional patterns and multiple planar mobile robots. Basically, an attractive vector field that guides the robots to a pattern is added to a field that is designed to avoid collisions among the robots. The main difference among these methods is the way such fields are computed. Chaimowicz et al. [9] compute the attractive field as the gradient of a function, which is constructed as an interpolation of several radial basis functions centered at samples of the desired pattern. This methodology, which allows the generation of generic shaped patterns, is used and extended in the present work to generate curves in two and three-dimensional spaces, as will be shown later in the paper. Hsieh and Kumar in [10] used a similar approach, but improved Chaimowicz's inter-robot collision solution. This allowed for proofs of convergence for the group. Pimenta et al [11] proposed the use of a numerical method to compute an electrostatic analogous field to attract the group of robots to the pattern. The main advantage of their approach is the possibility to add static, previously known obstacles in the workspace. Another contribution of [11] was the use of fluid models to control the inter-robot behavior.

Solutions to pattern tracking using vector fields are very similar to the ones adopted for pattern generation with multiple robots. In addition to the field that attracts the robot to the target curve, it is also necessary another vector field to make the robot traverse the curve. The computation of these fields and a method to compose them, allowing for proofs of convergence, are the main challenges to obtain good solutions. The composition of an attractive and a rotational field usually creates a stable limit cycle in the robot configuration space. Just to clarify the idea, we present in Figure 1 an example of a pattern in a two-dimensional space and a vector field that solves the tracking problem. This idea was used in [12], [13], [14], [15], [16], [17], [18], [19], [20], and [21].

Hsieh et al. [15] compute a dynamic vector field to control a team of planar holonomic mobile robots to converge to and circulate along the boundary of a desired static two-dimensional geometric pattern. The vector field is dynamic in the sense that it changes according to the robots' relative positions. This is necessary to guarantee collision avoidance. Ceccarelli et al. [18] also deal with planar ground robots. Although the pattern is constrained to be circular, proofs of convergence for a group of nonholonomic robots are presented.

V. M. Gonçalves, L. C. A. Pimenta, C. A. Maia, B. C. O. Dutra, and G. A. S. Pereira are with the Escola de Engenharia, Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, MG, 31270-901, Brazil. {mariano, lucpim, maia, gpereira}@cpdee.ufmg.br, brunocodutra@gmail.com

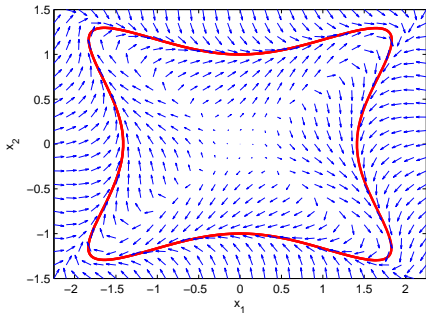


Fig. 1. Example of curve and vector field.

Several other papers on pattern tracking are motivated by the nature of the robot used in the application [16], [21], [19]. In the case of airplane based UAVs (Unmanned Air Vehicles), the robot must maintain a positive forward speed and cannot converge to a single point. Therefore, most of the tasks may be modeled as closed patterns tracking problems. For example, Frew et al. [16] controlled two UAVs to follow moving targets on the ground by modeling the task as the problem of tracking a loiter circular pattern centered at the target. Although the authors experimentally show that a vector field can be used to solve the problem, they did not present formal proofs that consider the time-varying nature of the pattern.

In some situations, traditional waypoint navigation problems may be also seen as a pattern tracking problem if the waypoints are considered to be sample points of a closed curve, as shown in [21]. In these cases, the main challenge is to construct a continuous vector field that guides the robot to follow an arbitrary sequence of waypoints. Using the methodology proposed by [19], for example, it is possible to generate fully continuous vector fields for any pattern that may be constructed by warping a circular curve. Since the warping technique proposed by the authors does not allow for tracking any arbitrary sequence of waypoints, they also propose a switching algorithm that generates discontinuous vector fields. Iscold et al. [21] compute a continuous vector field that allows a UAV to track an arbitrary sequence of planar waypoints. The authors construct a polygonal corridor that contains a set of waypoints and apply the methodology proposed in [4] to generate the field inside the corridor. The main drawbacks of the methodology are the field discontinuity at the border of the corridor and the difficulty to extend the vector field to three-dimensional workspaces. Actually, solutions that generate time-varying vector fields for pattern tracking in dimensions higher than two could not be found in the literature until the present date.

In the present work we address the problem of computing a vector field to control a single robot, represented by its configuration, to converge to and circulate along a given curve. Unlike previous works, we compute vector fields to deal with the general problem of time-varying patterns in n -dimensional spaces. To the best of the authors' knowledge, the methodology in the present paper is the first vector field approach that explicitly consider generic shaped time-varying patterns, defined in n -dimensional spaces, with convergence

proofs. The fact that the methodology is able to control robots with n degrees of freedom allows it to be used to control, not only mobile robots, but also manipulators performing repetitive tasks that can be represented by closed curves in their n -dimensional configuration spaces.

This paper extends our preliminary results presented in [22], in which we have shown an early version of the proposed vector field. This field is based on the sum of three terms: (i) convergence; (ii) circulation; and (iii) correction term. The first term guarantees that a kinematically controlled holonomic robot approaches the desired curve, the second one guarantees that the robot circulates the curve, and the third is a feedforward term that compensates for the time-varying nature of the curve. These components are based on $n - 1$ implicit functions that define the target curve. Unlike the preliminary version of the work in [22], in the present paper we have replaced the negative definite function used in the convergence term by a more intuitive potential function, which is built from the implicit functions that define the target curve. The mathematical proofs are detailed and presented in a more constructive way, and an extension of the methodology to second-order dynamic systems is also presented. Furthermore, we present a practical methodology to determine the implicit functions using a set of sample points of the target curve. This methodology may be used for static and time-varying periodic curves. Similar to previous works [23], [9], this technique is based on radial basis function interpolation. However, we use such an interpolation in spaces with dimensions higher than two and also in the context of time-varying boundaries. Finally, in this paper we discuss how the proposed methodology can be used to guide and control actual robots, and we illustrate the method with a differential drive mobile robot.

The paper is organized as follows. In Section II the problem is formulated. The proposed vector field methodology to solve the problem is presented in Section III. In this section we also present mathematical guarantees and the computational complexity of the method. In Section IV we present a practical numerical technique to construct the desired curves and discuss how the method can be applied to control actual robots. Conclusions and directions for future research are presented in Section V. In Appendix A we show an extension to our solution, which is initially proposed to single integrator robots, to second-order systems. Finally, in Appendix B we present mathematical results for a special class of implicit functions that shows that the only set of configurations which may cause our technique to fail is of measure zero.

II. PROBLEM DEFINITION

We consider a robot R represented by a configuration vector $\mathbf{q} = [x_1 \ x_2 \ \dots \ x_n]^T$ in the n -dimensional, obstacle-free configuration space $\mathcal{C} = \mathbb{R}^n$. We initially assume that the robot is modeled as a single integrator:

$$\dot{\mathbf{q}} = \mathbf{u}, \quad (1)$$

in which \mathbf{u} is the control input¹. Embedded in \mathcal{C} , we define a one-dimensional target curve $\mathcal{T}(t)$, where t is time, which

¹In Appendix A we present an extension for second-order systems.

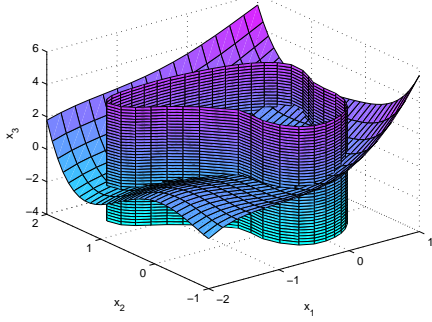


Fig. 2. Two-dimensional surfaces in \mathbb{R}^3 such that the intersection defines a one-dimensional closed curve

is expressed as the intersection of $n - 1$ surfaces of co-dimension 1. We assume that a $(n - 1)$ -dimensional surface is a level set of a function $\alpha : \mathbb{R}^n \mapsto \mathbb{R}$. In fact, $\mathcal{T}(t)$ is described as the set of points such that, for an appropriate set of functions α_i , $\alpha_i(x_1, x_2, \dots, x_n, t) = 0$ for $i = 1, 2, \dots, n - 1$ at those points. Figure 2 is an example of a one-dimensional curve in \mathbb{R}^3 that is the intersection of two two-dimensional, static surfaces $\alpha_1(x_1, x_2, x_3) = 0$ and $\alpha_2(x_1, x_2, x_3) = 0$. By considering these assumptions, the problem we are solving in this paper can be stated as:

Problem 1 *Let \mathcal{T} be a closed curve, which may be static or time-varying, defined in a n -dimensional space. Compute a static or time-varying vector field, \mathbf{h} , such that its integral curves asymptotically converge to and circulate along \mathcal{T} .*

More precisely, we can rewrite Problem 1 as: create a vector field $\mathbf{h}(\mathbf{q}) : \mathcal{C} \rightarrow T_{\mathbf{q}}(\mathcal{C})$, where $T_{\mathbf{q}}(\mathcal{C})$ is the tangent space of \mathcal{C} , such that if $\mathbf{u} = \mathbf{h}(\mathbf{q})$ then: (i) as $t \rightarrow \infty$, $\mathbf{q} \rightarrow \mathcal{T}(t)$; and (ii) once in \mathcal{T} this set is continuously traversed by \mathbf{q} in a fixed direction. In the case of static curves, statement (ii) is satisfied if $\dot{\mathbf{q}}$ is continuous and $\|\dot{\mathbf{q}}\| > 0$, $\forall t \geq 0$. For time-varying curves, it is satisfied if $\dot{\mathbf{q}}$ is continuous and $\|\mathbf{R}\| > \|\mathbf{v}_{\mathcal{T}}(\mathbf{q}, t)\|$, $\forall t \geq 0$, where \mathbf{R} is the component of $\dot{\mathbf{q}}$ tangent to the curve and $\mathbf{v}_{\mathcal{T}}(\mathbf{q}, t)$ is the tangential velocity of the curve at \mathbf{q} .

Before we proceed with the proposed vector field methodology, we provide formal definitions for the functions α_i and for the curve $\mathcal{T}(t)$.

Definition 1 *Let $\alpha_i(x_1, x_2, \dots, x_n, t) : \mathbb{R}^{n+1} \mapsto \mathbb{R}$, $i = 1, 2, 3, \dots, n - 1$ be functions with bounded second partial derivatives such that the set $\mathcal{T}(t)$, as given by Definition 2, is connected and one-dimensional $\forall t \geq 0$.*

Definition 2 *Let $\mathcal{T}(t) = \{[x_1 \ x_2 \ \dots \ x_n]^T \in \mathbb{R}^n | \alpha_i(x_1, x_2, \dots, x_n, t) = 0, \forall i \leq n - 1\}$ be the set of all points that lie in the intersection of the level sets $\alpha_i = 0$.*

The requirement for bounded second partial derivatives in Definition 1 will become clear later in our proofs. Notice that we define α_i functions such that the time-varying set $\mathcal{T}(t)$ is one-dimensional and connected for all $t \geq 0$. Connectedness is necessary to avoid disjoint closed curves.

III. METHODOLOGY

This section presents a solution to the problem defined in Section II. This problem can be decomposed into two subproblems: (i) convergence to the curve and (ii) circulation along it. Each of these subproblems will be solved separately using vector fields. In what follows, the vector field, which will be obtained as a composition of the individual solutions, will be shown to ensure both convergence and circulation.

A. Convergence to the curve

Let $V : \mathbb{R}^{n-1} \mapsto \mathbb{R}$ be a differentiable positive definite function. Since α_i is a function of x_1, x_2, \dots, x_n and t , $V(\alpha_1, \alpha_2, \dots, \alpha_{n-1})$ is also a function of these variables. Furthermore, since V is positive definite, $V = 0$ if and only if $\alpha_i = 0$, $\forall i$. Therefore, if we are able to set the function V to zero, then we can guarantee convergence to the curve. Note that

$$\frac{dV}{dt} = \nabla V^T \dot{\mathbf{q}} + \frac{\partial V}{\partial t}, \quad (2)$$

where ∇ is the gradient taken with respect to \mathbf{q} .

If \dot{V} is a negative definite function, then $V \rightarrow 0$. It is important to mention that, by Definition 1, the second partial derivatives of the functions α_i are bounded, which implies that the first partial derivatives are continuous. Also, V is differentiable, which means that both ∇V and $\partial V / \partial t$ are well defined. Let

$$\mathbf{u} = -G\nabla V + \mathbf{P}, \quad (3)$$

where \mathbf{P} is a vector to be defined and $G(x_1, x_2, \dots, x_n, t)$ is a non-negative scalar function. Vector \mathbf{u} can be understood as a combination of two terms. The first term is given by $-G\nabla V$ which is responsible for guiding the system to a minimum of V , and provided that certain conditions are ensured the only possible minimum is $V = 0$. As previously mentioned, since V is positive definite this minimum is only reached if $\alpha_i = 0$, $\forall i$. Therefore, this first term drives the system toward the desired curve. The second term is a vector \mathbf{P} which will be designed to compensate for the possible time-varying nature of the curve. It should be clear that this term is not necessary when the curve is static.

According to the model in (1) and substituting (3) in (2) we obtain

$$\frac{dV}{dt} = -G\|\nabla V\|^2 + \nabla V^T \mathbf{P} + \frac{\partial V}{\partial t}. \quad (4)$$

If

$$\nabla V^T \mathbf{P} + \frac{\partial V}{\partial t} = 0, \quad (5)$$

then $\dot{V} \leq 0$. Using the chain rule

$$\nabla V = \sum_{i=1}^{n-1} \frac{\partial V}{\partial \alpha_i} \nabla \alpha_i, \quad (6)$$

$$\frac{\partial V}{\partial t} = \sum_{i=1}^{n-1} \frac{\partial V}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial t}. \quad (7)$$

By replacing these two expressions in (5) we have:

$$\sum_{i=1}^{n-1} \frac{\partial V}{\partial \alpha_i} \left(\nabla \alpha_i^T \mathbf{P} + \frac{\partial \alpha_i}{\partial t} \right) = 0. \quad (8)$$

Therefore, in order to satisfy (8), it is sufficient that

$$\nabla\alpha_i^T \mathbf{P} = -\frac{\partial\alpha_i}{\partial t}, \quad (9)$$

for $i = 1, 2, \dots, n-1$. To rewrite (9) in a compact form, the following definition is needed.

Definition 3 Let M_* be the matrix in $\mathbb{R}^{(n-1) \times n}$ such that the i^{th} row is given by the vector $\nabla\alpha_i^T$ for $i = 1, 2, \dots, n-1$. Also, let \mathbf{a}_* be the column vector in \mathbb{R}^{n-1} such that the i^{th} row is given by $\partial\alpha_i/\partial t$ for $i = 1, 2, \dots, n-1$.

Using Definition 3 we can rewrite the set of equations in (9) as:

$$M_* \mathbf{P} = -\mathbf{a}_*. \quad (10)$$

Definition 4 Let $\mathcal{U}(t)$ be the set of points $[x_1 \ x_2 \ \dots \ x_n] \in \mathbb{R}^n$ such that the vectors $\nabla\alpha_i$'s are linearly dependent.

In Appendix B it is proved that some reasonable assumptions are sufficient to guarantee that $\mathcal{U}(t)$ is of measure zero on \mathbb{R}^n . We also want this set to be repulsive according to the next definition.

Definition 5 A time-varying set of points, $\mathcal{S}(t)$, of a dynamical system $\dot{\mathbf{q}} = \mathbf{h}(\mathbf{q}, t)$ is said to be a repulsive set if there exists a neighborhood $\mathcal{N}(\mathcal{S})$ such that for all $\mathbf{q} \in \mathcal{N}$ we have $\dot{D} > 0$ for all t , where D is the distance between \mathbf{q} and \mathcal{S} .

Assuming that the set $\mathcal{U}(t)$ is repulsive for all $t > 0$, the existence of a solution \mathbf{P} in Equation (10) is independent of the vector \mathbf{a}_* . We will explicitly write a solution later. At this point, we will assume that it is possible to obtain such a \mathbf{P} , and then Equation (4) becomes:

$$\frac{dV}{dt} = -G \|\nabla V\|^2. \quad (11)$$

We can see that if G is null on the points such that $\nabla V = \mathbf{0}$ then the qualitative behavior of (11) is unchanged. Thus, G is formalized as follows:

Definition 6 Let $G(x_1, x_2, \dots, x_n, t) : \mathbb{R}^{n+1} \mapsto \mathbb{R}$ be a non-negative function with bounded partial derivatives. Besides, it can only vanish in configurations where $\nabla V = \mathbf{0}$.

Now, it is important to analyze the term $\|\nabla V\|^2$. This term is null if and only if $\nabla V = \mathbf{0}$. For a differentiable positive definite function V , the partial derivatives $\partial V/\partial\alpha_i = 0$ vanishes at the arguments $\alpha_1 = \alpha_2 = \dots = \alpha_{n-1} = 0$, which means that $\nabla V = \mathbf{0}$ in $\mathcal{T}(t)$. By (6), we can see that if at least one of the components $\partial V/\partial\alpha_i$ is non null for a point \mathbf{q} at time t and $\nabla V = \mathbf{0}$, then the vectors $\nabla\alpha_i$ are linearly dependent. Therefore, these points form a subset of the set $\mathcal{U}(t)$. Without further restrictions on V , there are also remaining points such that $\nabla V = \mathbf{0}$ that are neither in $\mathcal{T}(t)$ nor in $\mathcal{U}(t)$: the points where all components $\partial V/\partial\alpha_i$ vanish and are not in $\mathcal{T}(t)$. The following definition of V ensure that this set is empty.

Definition 7 Let $V : \mathbb{R}^{n-1} \mapsto \mathbb{R}$ be a positive definite function with bounded second order partial derivatives such that its gradient is null only at the origin. We will refer to this function V along the text as a potential function.

Provided that the set $\mathcal{U}(t)$ is repulsive, the right side of Equation (3) solves the problem of convergence. It is important to mention that an explicit expression for \mathbf{P} will be shown in Subsection III-C.

B. Circulation along the curve

Suppose that \mathbf{q} has already converged to the set $\mathcal{T}(t)$ for a given t . Therefore, $\nabla V = \mathbf{0}$. In order to make the system to traverse the curve, we add a term \mathbf{R} to (3):

$$\mathbf{u} = \mathbf{R} + \mathbf{P}. \quad (12)$$

To maintain the configuration on the curve, moving along it, it is important to ensure that $\|\mathbf{R}\| \neq 0$ (see Section II) and $\dot{\alpha}_i = 0$ for $i = 1, 2, \dots, n-1$. Observe that

$$\frac{d\alpha_i}{dt} = \nabla\alpha_i^T \dot{\mathbf{q}} + \frac{\partial\alpha_i}{\partial t}. \quad (13)$$

By remembering that $\dot{\mathbf{q}} = \mathbf{u}$, and using (12) and (9) in (13) we get

$$\nabla\alpha_i^T \mathbf{R} = 0, \quad (14)$$

for $i = 1, 2, \dots, n-1$, or simply

$$M_* \mathbf{R} = \mathbf{0}. \quad (15)$$

If the vectors $\nabla\alpha_i$ are linearly independent, then there is only one vector (up to a scalar factor) in \mathbb{R}^n that satisfies (15). This motivates the next definition.

Definition 8 Let \mathbf{v}_i , $i = 1, 2, 3, \dots, n-1$, and $n \geq 2$ be row vectors in \mathbb{R}^n . The wedge product [24] $\wedge_{i=1}^{n-1} \mathbf{v}_i$ is the vector in \mathbb{R}^n such that the i^{th} element is given by the cofactor of n^{th} row and i^{th} column of the matrix such that the i^{th} row is given by the vector \mathbf{v}_i for $i = 1, 2, 3, \dots, n-1$ (since the last row is unnecessary, it is left undefined).

The definition above can be seen as an extension of the cross product in \mathbb{R}^3 to \mathbb{R}^n . Note that, if $n = 3$ this definition reduces to the usual cross product. The resulting vector is orthogonal to each \mathbf{v}_i . Besides, this vector is null whenever two (or more) vectors are linearly dependent. In this paper, if $n = 3$ we will write $\mathbf{v}_1 \times \mathbf{v}_2$ instead of $\wedge_{i=1}^2 \mathbf{v}_i$. It is interesting to observe that Definition 4 can be rewritten using the wedge product: the set $\mathcal{U}(t)$ is the set of points such that $\wedge_{i=1}^{n-1} \nabla\alpha_i = \mathbf{0}$. In Appendix B, we use this observation to derive a sufficient condition for $\mathcal{U}(t)$ to be a set of measure zero. Basically, it is shown that if the functions α_i are real analytic and their respective gradients are not zero almost everywhere for all $t \geq 0$, then the set $\mathcal{U}(t)$ has measure zero.

It becomes clear that if

$$\mathbf{R} = H \wedge_{i=1}^{n-1} \nabla\alpha_i, \quad (16)$$

in which $H(x_1, x_2, \dots, x_n, t)$ is a scalar function according to the following definition, then (14) holds for $i = 1, 2, \dots, n-1$.

Definition 9 We denote by $H(x_1, x_2, \dots, x_n, t) : \mathbb{R}^{n+1} \mapsto \mathbb{R}$ a continuous, strictly positive or negative function at $\mathcal{T}(t)$.

Observe that each point of a time-varying curve moves according to a velocity, which, in general, may have a component tangent to the curve. We will choose in the next subsection a particular vector \mathbf{P} , which is orthogonal to \mathbf{R} . Thus, the only term that guides the system in the tangent direction of $\mathcal{T}(t)$ is \mathbf{R} . Therefore, circulation can always be ensured by choosing a proper H which makes $\|\mathbf{R}\|$ larger than the norm of any tangent velocity component of the curve.

It is important to mention the need of $\mathcal{T}(t) \cap \mathcal{U}(t) = \emptyset$. If the intersection is not empty, *i.e.* there are points on the target curve such that the gradients are linearly dependent, then the term \mathbf{R} vanishes.

As previously pointed out, not all choices of functions α_i are such that the set \mathcal{T} , *i. e.* $\alpha_i = 0 \forall i$, is one-dimensional for all $t > 0$. Let \mathbf{q}_f be a point in this set. Then, locally and for a given t , all the surfaces $\alpha_i = 0$ are approximately a $n - 1$ dimensional plane near that point, provided that the functions α_i are differentiable in their spatial arguments x_i . For \mathbf{q} sufficiently close to \mathbf{q}_f we have:

$$\nabla \alpha_i^T|_{\mathbf{q} \rightarrow \mathbf{q}_f} (\mathbf{q} - \mathbf{q}_f) = 0, \quad (17)$$

for $i = 1, 2, \dots, n - 1$. This equation corresponds to the first order Taylor expansion of α_i close to \mathbf{q}_f . A sufficient condition for \mathcal{T} to be one dimensional is that the $n - 1$ vectors $\nabla \alpha_i|_{\mathbf{q} \rightarrow \mathbf{q}_f}$ are linearly independent, since the set of points \mathbf{q} that satisfies (17), $\forall i$, will be one dimensional.

Observe that it is sufficient that $\wedge_{i=1}^{n-1} \nabla \alpha_i \neq \mathbf{0}$ for the intersection set to be one-dimensional. However, this is not necessary: if $n = 3$ and $\alpha_1 = x_1^2 + x_2^2 + x_3^2 - 1$ and $\alpha_2 = x_1^2 + x_2^2 - 1 = 0$, for example, the intersection set is one dimensional (the circumference $x_1^2 + x_2^2 - 1 = 0$ in the x_1x_2 plane) but the wedge product is null on the curve. The set of functions α_i that provides curves with this behavior (one-dimensional but with null wedge product on the curve) must be avoided since in this case circulation will be compromised.

Provided that the set $\mathcal{U}(t)$ is repulsive and that $\mathcal{T}(t) \cap \mathcal{U}(t) = \emptyset$, the control law in (12) solves the problem of circulation. In the next subsection convergence and circulation will be ensured.

C. Circulation and Convergence

This subsection starts with a definition:

Definition 10 Let M be the matrix in $\mathbb{R}^{n \times n}$ such that the first $n - 1$ lines correspond to matrix M_* and the last line is the vector $(\wedge_{i=1}^{n-1} \nabla \alpha_i)^T$. Also, let \mathbf{a} be the vector in \mathbb{R}^n such that the first $n - 1$ lines correspond to vector \mathbf{a}_* and the last line is 0.

Note that matrix M is invertible if and only if \mathbf{q} is not in $\mathcal{U}(t)$. Thus, if \mathbf{P} satisfies

$$M\mathbf{P} = -\mathbf{a}, \quad (18)$$

then it clearly satisfies (10). Furthermore, the last line in (18) means that $\wedge_{i=1}^{n-1} \nabla \alpha_i$ and \mathbf{P} are orthogonal. Therefore, \mathbf{R} (as

in (16)) and \mathbf{P} are also orthogonal and the sum $\mathbf{R} + \mathbf{P}$ is null only if both terms are null. Now, we can write \mathbf{P} provided that the system is not in $\mathcal{U}(t)$:

$$\mathbf{P} = -M^{-1}\mathbf{a}.$$

The term \mathbf{P} is essential to guarantee the convergence to time-varying curves, as will be discussed in the next subsection.

Now, we will merge the right sides of equations (3) and (12):

$$\mathbf{u} = -G\nabla V + H(\wedge_{i=1}^{n-1} \nabla \alpha_i) - M^{-1}\mathbf{a}. \quad (19)$$

The resulting vector field solves both the problem of circulation and of convergence, provided that the set $\mathcal{U}(t)$ is repulsive and $\mathcal{T}(t) \cap \mathcal{U}(t) = \emptyset$. This will be proved in the next theorem by using the Lyapunov-like lemma presented in [25]. This lemma follows directly from the well-known Barbalat's lemma.

Theorem 1 Consider the nonautonomous dynamical system defined by equations (1) and (19). Assume that the set $\mathcal{U}(t)$ is repulsive and that $\mathcal{T}(t) \cap \mathcal{U}(t) = \emptyset$ for all t . The system converges asymptotically to the set $\mathcal{T}(t)$ for any initial condition ($t = 0$) \mathbf{q}_0 that is not in $\mathcal{U}(0)$. In addition, it is always possible to ensure circulation along $\mathcal{T}(t)$ with a given fixed direction. Moreover, $\dot{\mathbf{q}}$ is continuous.

Proof: Note that $\dot{\mathbf{q}}$ is continuous for any t due to the differentiability or continuity of the involved functions and that, by hypothesis, the points in the set $\mathcal{U}(t)$ are repulsive.

Function V is positive definite and, therefore, lower bounded. It can be verified that

$$\frac{dV}{dt} = -G\|\nabla V\|^2 + H\nabla V^T(\wedge_{i=1}^{n-1} \nabla \alpha_i).$$

Using (6) and the fact that $\wedge_{i=1}^{n-1} \nabla \alpha_i$ is orthogonal to each $\nabla \alpha_i$ we obtain:

$$\frac{dV}{dt} = -G\|\nabla V\|^2.$$

By the assumptions of bounded second derivatives of V and α_i and bounded first derivative of G , we can check that d^2V/dt^2 is bounded. Thus, the Lyapunov-like lemma in [25] is applicable. Therefore, $dV/dt \rightarrow 0$. Provided that the set $\mathcal{U}(t)$ is repulsive, $V \rightarrow 0$ and the system converges asymptotically to the desired curve.

Once in the curve, the term $-G\nabla V$ vanishes and the system input is given by (12). By the discussion in Subsection III-B, it is always possible to choose H such that circulation is ensured. The sign of H in the desired curve determines the direction of circulation. ■

D. Numerical Example

In this subsection we illustrate the result of Theorem 1 and discuss the importance of the correction term in the proposed control law by means of a numerical example. We consider convergence and circulation of a time-varying curve in \mathbb{R}^3 . Let:

$$\begin{aligned} \alpha_1(x_1, x_2, x_3, t) &= x_1^2 + x_2^2 - 1, \\ \alpha_2(x_1, x_2, x_3, t) &= x_3 - \sin(\gamma t), \end{aligned} \quad (20)$$

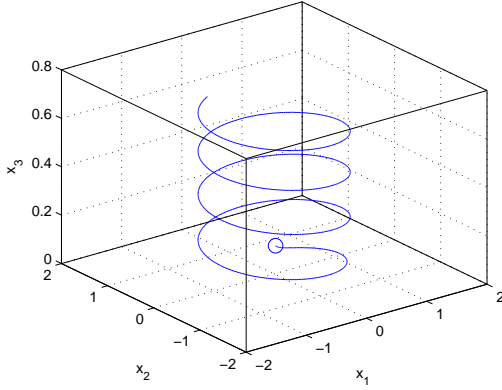


Fig. 3. Simulation for an initial condition $\mathbf{q}_0 = [0.1 \ 0.1 \ 0.1]^T$.

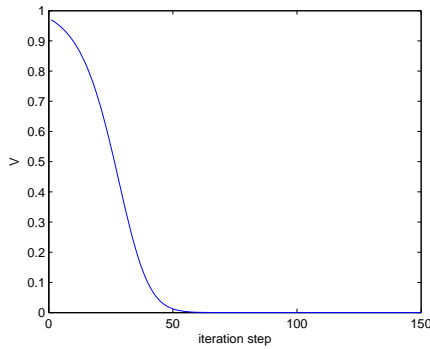


Fig. 4. Evolution of the potential function for the simulation in Figure 3.

in which $\gamma \neq 0$. The set $\mathcal{T}(t)$ is one dimensional: a circumference parallel to the x_1x_2 plane with unitary radius and center at the x_3 axis that changes between $x_3 = 1$ and $x_3 = -1$ (according to $x_3 = \sin(\gamma t)$). We first need to find the set $\mathcal{U}(t)$. The wedge product between $\nabla\alpha_1$ and $\nabla\alpha_2$ (cross product, for $n = 3$) is given by

$$\nabla\alpha_1 \times \nabla\alpha_2 = [2x_2 \ -2x_1 \ 0]^T.$$

Therefore, the set $\mathcal{U}(t)$ ($\nabla\alpha_1 \times \nabla\alpha_2 = \mathbf{0}$) is the set $\{(x_1, x_2, x_3) \in \mathbb{R}^3 | x_1 = x_2 = 0\}$ (the x_3 axis). Furthermore, it is easy to see that the cross product is non null on the desired curve. Now, it is necessary to select a potential function $V(\alpha_1, \alpha_2)$. We choose $V = \alpha_1^2 + \alpha_2^2$. As its gradient is given by $[2\alpha_1 \ 2\alpha_2]^T$, it is null only if $\alpha_1 = \alpha_2 = 0$. Moreover, all the requirements in Definition 7 are satisfied.

Now, we will prove that the set $\mathcal{U}(t)$ is repulsive for all t . This can be done by checking that if $D(x_1, x_2, x_3, t)$ denotes the distance between \mathbf{q} at time t and the set $\mathcal{U}(t)$, then $D^2 = \alpha_1 + 1$. The derivative of D^2 with respect to time is given by:

$$\frac{dD^2}{dt} = \nabla\alpha_1^T \dot{\mathbf{q}} = -G \frac{\partial V}{\partial \alpha_1} \|\nabla\alpha_1\|^2 - \nabla\alpha_1^T M^{-1} \mathbf{a}, \quad (21)$$

since $\nabla\alpha_1$ and $\nabla\alpha_2$ are orthogonal. Technically, expression (21) is not defined in the set $\mathcal{U}(t)$ since $M^{-1}\mathbf{a}$ is undefined in this set. Therefore, we could worry about points that are very close to $\mathcal{U}(t)$. As $\mathbf{P} = -M^{-1}\mathbf{a}$ and the first row

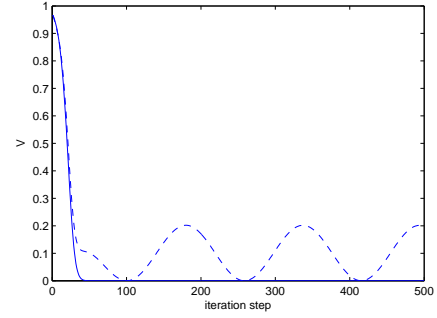


Fig. 5. Evolution of the potential function with (solid line) and without (dashed line) the time-varying correction term \mathbf{P} . For the latter, notice that the system oscillates near the target.

of (18) means that $-\nabla\alpha_1^T M^{-1} \mathbf{a} = \nabla\alpha_1^T \mathbf{P} = -\partial\alpha_1/\partial t = 0$ we obtain

$$\frac{dD^2}{dt} = \nabla\alpha_1^T \dot{\mathbf{q}} = -G \frac{\partial V}{\partial \alpha_1} \|\nabla\alpha_1\|^2 = -2G\alpha_1 \|\nabla\alpha_1\|^2. \quad (22)$$

We can notice that in order \mathbf{q} to be close to $\mathcal{U}(t)$, it is necessary that $\alpha_1 < 0$. Thus, expression (22) is positive near $\mathcal{U}(t)$ and this set is repulsive according to Definition 5. In this case, $\mathcal{N}(\mathcal{U}(t))$ is the set $\{(x_1, x_2, x_3) \in \mathbb{R}^3 | \alpha_1(x_1, x_2, x_3, t) < 0\}$.

Figure 3 shows a simulation for an initial condition $\mathbf{q}_0 = [0.1 \ 0.1 \ 0.1]^T$. Figure 4 shows the evolution of the potential function in time. It is clear that $V \rightarrow 0$ as $t \rightarrow \infty$. For this simulation we chose $G = H = 1$ and $\gamma = 0.05$.

The term \mathbf{P} , discussed in the previous subsections, is essential to achieve convergence if the curve is time-varying. In this case, the absence of this term may cause the system to converge with a fluctuating error in the best case, and to diverge in the worst scenario.

Consider the given numerical example, but now with $\gamma = 1$. As we can infer by Figure 5, that shows the plot of V for the controller with and without \mathbf{P} , in the latter the system oscillates close to the desired curve but does not converge. This phenomenon has an intuitive explanation. Without the correction term, the only one that performs the convergence task is $-G\nabla V$. When \mathbf{q} approaches the target, this term tends to vanish and therefore the convergence rate decreases, but the curve keep varying, what increases the error between \mathbf{q} and the target as the controller convergence term is too slow to react. This prompts the term to increase again, and so on, creating the oscillating behavior. In some cases, however, the system cannot even approach the time-varying curve because it may be too fast to be tracked without the correction term.

The behavior of the system when the term is not used is, of course, dependent on the choice of the parameters. In order to show the importance of the term, we used $\gamma = 1$ in (20) since it is almost impossible to see the difference between the performance of the controller with and without the term (the oscillation in V is very small) with $\gamma = 0.05$. This is due to the fact that the curve is too slow compared to our choice of G and V , and so \mathbf{P} is not so important as when $\gamma = 1$.

A theoretical analysis of the correction term is presented in [26].

E. Computational Complexity

The computational implementation of the method proposed in this paper relies on the user ability to obtain $n - 1$ functions with n arguments that define the target curve in \mathbb{R}^n . For a special case, an algorithm for constructing these functions from sample points of the target curve is presented in Section IV-A. We do not take this into account in the computational complexity analysis presented in this section since it is an off-line pre-processing step.

In real time, the robot must estimate its configuration \mathbf{q} and use it to evaluate each of the $n - 1$ functions $\partial V / \partial \alpha_i$ in order to compute its control law. This is a $O(n)$ operation. The gradient of each function α_i at \mathbf{q} is also necessary. Each gradient has n coordinates and therefore a $O(n^2)$ algorithm is used. For a static curve, given the value of those functions, the computation of the convergence term is given by the summation in Equation (6) which can be done in $O(n^2)$. The circulation term corresponds to the computation of the wedge product of the $n - 1$ gradient vectors of functions α_i . This requires a $O(n^4)$ algorithm since it is necessary to compute n determinants of order $n - 1$, which are usually computed by an algorithm based on LU decomposition consuming $O(n^3)$ operations. Therefore, for static curves the proposed algorithm is $O(n^4)$.

For time-varying curves, it is also necessary to compute the time derivative of the $n - 1$ α_i functions, which is $O(n)$, the inverse of the $n \times n$ matrix M , and the product $M^{-1}\mathbf{a}$. A common algorithm for inverse computation has complexity $O(n^3)$. The product $M^{-1}\mathbf{a}$ can be done in $O(n^2)$.

Thus, for either static or time varying curves, the complexity of the algorithm is $O(n^4)$. It is important to emphasize that for a typical n , such a computational cost is not an issue.

IV. REAL WORLD IMPLEMENTATION

A. Construction of Curves

The proposed methodology relies on our ability to determine a set of distinct α_i functions, such that the intersection of the level surfaces $\alpha_i = 0$ is the desired curve. In this subsection we present a technique for constructing such intersections based on a set of interpolation points. First we address static boundaries and second we address time-varying boundaries.

1) *Static Boundaries*: Similarly to [23] and [9], we use radial basis functions to interpolate a set of samples of a given curve. In fact, in this paper we extend the previous works to dimensions higher than two. We will consider curves in \mathbb{R}^3 , but our ideas can be further extended to higher dimensions as we will discuss at the end of this subsection.

Definition 11 A radial basis function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is any function that depends only on the distance between \mathbf{q} and a fixed center \mathbf{q}_c : $f(\mathbf{q}, \mathbf{q}_c) = F(\|\mathbf{q} - \mathbf{q}_c\|)$.

The next definition will be useful in our main assumption.

Definition 12 Let $\Pi(\mathbf{q}, \mathcal{P})$ be the orthogonal projection of the point \mathbf{q} onto the plane \mathcal{P} .

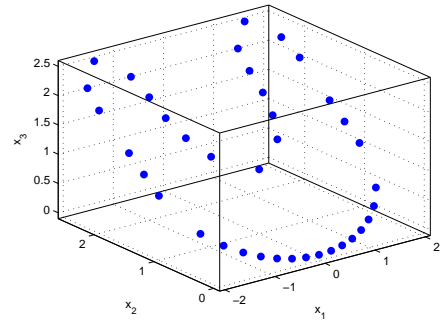


Fig. 6. Sample points of a “U” shaped curve.

Assumption 1 The desired curve \mathcal{T} is such that we can always find a plane in which $\Pi(\mathbf{q}_1, \mathcal{P}) \neq \Pi(\mathbf{q}_2, \mathcal{P})$, for every two points $\mathbf{q}_1 \in \mathcal{T}$ and $\mathbf{q}_2 \in \mathcal{T}$.

For the sake of simplicity, we will assume that the projection plane \mathcal{P} is given by $x_3 = 0$. Note that this is not a constraint, since, in principle, we can create our own orthogonal coordinates x_1, x_2, x_3 such that our plane is given by $x_3 = 0$.

Suppose we sample n_p points of the curve, $\tilde{\mathbf{q}}_i = [\tilde{x}_{1,i} \ \tilde{x}_{2,i} \ \tilde{x}_{3,i}]^T$. To create α_1 we use the orthogonal projections $\Pi(\tilde{\mathbf{q}}_i, \mathcal{P})$, where \mathcal{P} is the plane given by $x_3 = 0$. Consider the vectors $\mathbf{q}_* = [x_1 \ x_2 \ 0]^T$ and $\tilde{\mathbf{q}}_{*i} = [\tilde{x}_{1,i} \ \tilde{x}_{2,i} \ 0]^T$. Function α_1 is given by:

$$\alpha_1(x_1, x_2, x_3) = -1 + \sum_{k=1}^{n_p} \omega_k f(\mathbf{q}_*, \tilde{\mathbf{q}}_{*k}), \quad (23)$$

where ω_k is a scalar weight. Since the function must vanish on the desired curve, we define the constraint $\alpha_1(\tilde{\mathbf{q}}_{*i}) = 0$. We can find the weights w_k by solving the associated linear system. Observe that we have a square matrix with dimension n_p . It is important to mention that α_1 does not depend on x_3 .

Similarly, the function α_2 is determined by:

$$\alpha_2(x_1, x_2, x_3) = x_3 - \sum_{k=1}^{n_p} \xi_k f(\mathbf{q}_*, \tilde{\mathbf{q}}_{*k}). \quad (24)$$

To compute the weights ξ_i we define the constraint $\alpha_2(\tilde{\mathbf{q}}_{*i}) = 0$. Equation (24) explains the need for Assumption 1, since the summation in the right hand side of this equation can be seen as an interpolation function to fit x_3 .

The two functions α_1 and α_2 are necessarily distinct, since α_2 depends on x_3 and α_1 does not. By construction, the gradients $\nabla \alpha_1$ and $\nabla \alpha_2$ are linearly dependent if and only if $\nabla \alpha_1$ is null. Moreover, the intersection of the curves $\alpha_1 = 0$ and $\alpha_2 = 0$ results in a curve that approximates the desired one with accuracy dependent on the number of samples n_p .

As an illustration of the methodology, a “U” shaped curve with $n_p = 38$ is presented in Figure 6.

In order to create α_1 , it is sometimes necessary to use additional points other than those sampled. In the example of the “U” shaped curve, Figure 7 shows the projection on the plane $x_3 = 0$ with one additional point out of the curve. For this point we set the constraint $\alpha_1 = -1$. This is necessary since constructing the curve using only the sample points

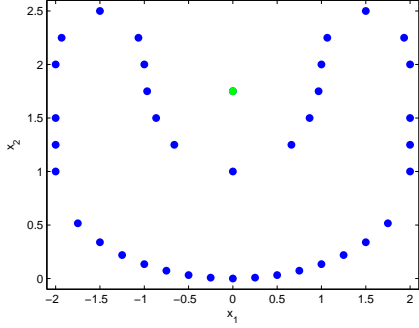


Fig. 7. Projection of the sample points in Figure 6 on the plane $x_3 = 0$, with one additional point out of the curve.

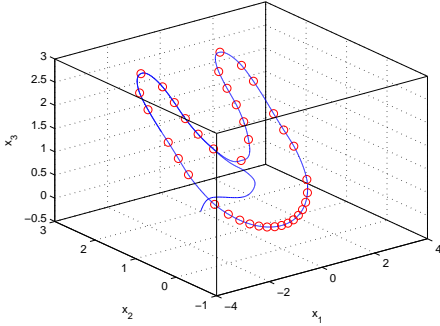


Fig. 8. Convergence to the target curve determined by the sample points in Figure 6 with initial condition given by $x_1(0) = 0$, $x_2(0) = 2$, and $x_3(0) = -0.5$.

creates a pathological α_1 such that the level surface $\alpha_1 = 0$ is disconnected. This yields in two curves: one being the desired one and the other being an undesired ellipse that appears in the middle of the outer region of the “U”. The negative constraint forces the function to be negative around this region eliminating the undesired curve.

However, it should be noticed that placing a negative constraint to eliminate the given undesired curve could, in principle, create another one in another region. It should then be clear that the proposed method is not completely automatic. One must properly tune the method for a given application.

To create α_1 and α_2 we selected the radial basis function $F(u) = u \ln(u)$ (as in [23]) for $u \neq 0$ and 0 for $u = 0$. Note that in this case, the vector $\nabla F(\|\mathbf{q} - \mathbf{q}_i\|)$ is undefined when $\mathbf{q} = \mathbf{q}_i$. However, the limit of this vector when $\mathbf{q} \rightarrow \mathbf{q}_i$ exists and is equal to $\mathbf{0}$. Therefore, we just define $\nabla F(\|\mathbf{q} - \mathbf{q}_i\|) = \mathbf{0}$ when $\mathbf{q} = \mathbf{q}_i$.

Figure 8 shows the evolution of a robot configuration using the computed α_1 and α_2 for an initial configuration given by $\mathbf{q}_0 = [0 \ 2 \ -0.5]^T$. For this simulation we chose to use $V(\alpha_1, \alpha_2) = 5(\alpha_1^2 + \alpha_2^2)^{1.05}$ and $G = H = 1$.

This method can be extended to \mathbb{R}^n if we assume the following: there is a plane \mathcal{P}_n , that we will choose without loss of generality $x_n = 0$, such that the projection of the points $\tilde{\mathbf{q}}_i$ on this plane is unique. Then, again, we assume that this projected points can be uniquely projected on another plane \mathcal{P}_{n-1} that is orthogonal to \mathcal{P}_n , which we will choose without

loss of generality as $x_{n-1} = 0$.

We proceed then with these assumptions and projections until a collection of two dimensional points is reached. We then create two orthogonal coordinates x_1 and x_2 ($n-2$ steps are necessary). Let call the i -th sampled point on the $(n-p)$ -th step $\tilde{\mathbf{q}}_{*i}^p$ and $\mathbf{q}_*^p = [x_1 \ x_2 \ \dots \ x_p \ 0 \ \dots \ 0]^T$.

Using $\tilde{\mathbf{q}}_{*i}^2$, we then construct α_1 using Equation (23), so that α_1 only depends on x_1 and x_2 . From now, we use constructions similar to (24) using the points $\tilde{\mathbf{q}}_{*i}^p$, to construct α_p , $p = 2, 3, \dots, n-1$. So,

$$\alpha_p(x_1, x_2, \dots, x_n) = x_{p+1} - \sum_{k=1}^{n_p} \xi_k f(\mathbf{q}_*^p, \tilde{\mathbf{q}}_{*k}^p).$$

that depends only on x_1, x_2, \dots, x_{p+1} . This construction guarantees that the gradients $\nabla \alpha_i$ are linearly dependent if and only if $\nabla \alpha_1$ is null.

2) *Time-Varying Boundaries*: If the desired time-varying curve can be obtained by trivial operations on a fixed curve, such as rotating, translating, and expanding (shrinking) as the time pass, functions α_1 and α_2 could be obtained via their fixed counterparts. If not, another sampling process is necessary.

Suppose we sample the desired curve in time, at t_1, t_2, \dots, t_m , and that we want the curve to be periodic with period T , so, $t_1, t_2, \dots, t_m < T$. Each of these fixed curves could be constructed by the sampling process described in the previous subsection. Therefore, we obtain the sample functions $\tilde{\alpha}_i(x_1, x_2, x_3, t_1), \dots, \tilde{\alpha}_i(x_1, x_2, x_3, t_m)$ for $i = 1, 2$ (in \mathbb{R}^3). In order to construct the desired curve we can search for a function $\alpha_{i,complex}$ of the form:

$$\alpha_{i,complex}(x_1, x_2, x_3, t) = \sum_{k=1}^m \chi_{i,k}(x_1, x_2, x_3) L_k(e^{\frac{2\pi}{T}jt}),$$

where

$$L_i(e^{\frac{2\pi}{T}jt}) = \prod_{k=1, k \neq i}^m (e^{\frac{2\pi}{T}jt} - e^{\frac{2\pi}{T}jt_k}),$$

and $j^2 = -1$. This is similar to interpolating using a Lagrange polynomial. We set that at $t = t_k$, $\alpha_{i,complex}(x_1, x_2, x_3, t_k) = \tilde{\alpha}_i(x_1, x_2, x_3, t_k)$, and as $L_i(e^{\frac{2\pi}{T}jt_k}) = 0$ for all $i \neq k$ we obtain the weight functions $\chi_{i,k}$ in a closed form:

$$\chi_{i,k}(x_1, x_2, x_3) = \frac{\tilde{\alpha}_i(x_1, x_2, x_3, t_k)}{L_k(e^{\frac{2\pi}{T}jt_k})}.$$

By extracting the real part $\alpha_i = \Re\{\alpha_{i,complex}\}$, the resultant function α_i will be equal to the samples at the appropriate instants t_i and will be periodic with period T .

To illustrate the methodology, a time-varying curve was created, which was obtained by rotating the sampled curve on Figure 6 around the x_1 -axis. The curve periodically oscillates between -45 degrees and its 45 degrees rotated counterpart. Figure 9 depicts snapshots of the simulation where the largest circle represent the current robot configuration. We used the same V , G , and H functions of the previous simulation.

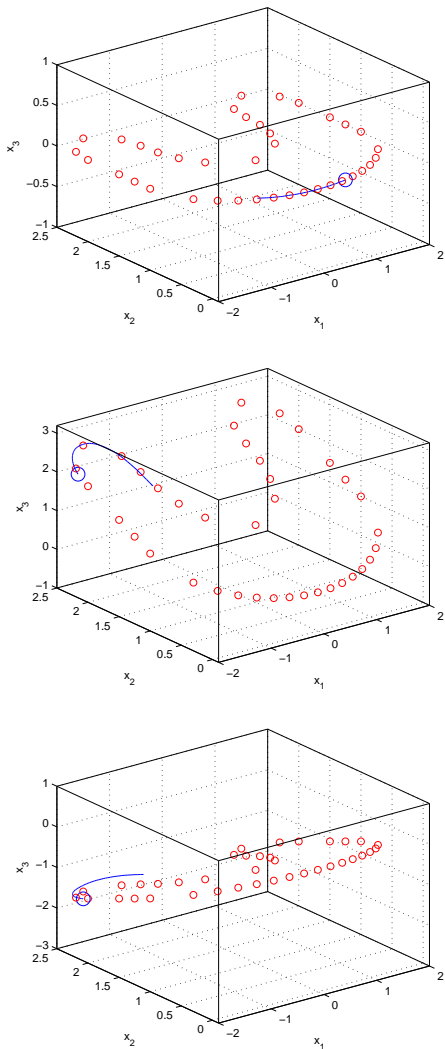


Fig. 9. Three snapshots of a simulation with a time-varying field constructed by a set of sample points of the desired curve. The robot configuration is represented by the largest circle.

B. Actual robots implementation

The methodology previously explained assumes holonomic robots represented by their exact configuration \mathbf{q} . Therefore, this theory can be directly applied to control a holonomic manipulator with n degrees of freedom. By using absolute sensors at the manipulator joints, the configuration, \mathbf{q} , can be obtained accurately. Probably, the biggest issue in this case would be the definition of the implicit functions in \mathbb{R}^n .

On the other hand, most of the commercially available mobile robot platforms are subject to kinematic constraints and use noisy sensor measurements for localization. Therefore, it is natural to ask if the proposed methodology can still be applied in the case of actual mobile robots (e.g. differential drive robots) navigating based on an estimated pose $\hat{\mathbf{q}}$.

The estimation of \mathbf{q} is a difficult problem in mobile robotics. Several good solutions have already been proposed for indoor localization [27], but accurate localization in outdoor and unstructured environments is still a challenge. However, an important advantage of vector field based approaches is the



Fig. 10. Picture of the robot used in the experiments.

fact that they are not too sensitive to small localization errors. Even in the presence of such errors, the robot will keep moving in the right direction (or very close to it).

In order to consider differential drive robots, such as the one used in the experiments presented here, the following model may be considered:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (25)$$

where the model inputs v and ω are the robot's linear and angular velocities, respectively, and θ is the robot orientation. A low level controller based on static feedback linearization [25], for example, could be used to enforce that the robot follows the desired vector field. We are interested in controlling the robot position $[x, y]^T$, but, in this case, it is not possible to directly use the model in (25) to compute the robots inputs, because the angular velocity does not appear explicitly in the equations of \dot{x} and \dot{y} . To circumvent this problem, we redefine the system output as $[x_d, y_d]^T = [x + d \cos(\theta), y + d \sin(\theta)]^T$, which corresponds to the position of the point $[d, 0]^T$ in the robot frame. This frame is attached to the robot center of mass and its x axis is parallel to the robot longitudinal axis point to the front. Therefore, the robot may be controlled by using $[\dot{x}_d, \dot{y}_d]^T = \mathbf{u}(\mathbf{q})$ which implies that:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\frac{\sin \theta}{d} & \frac{\cos \theta}{d} \end{bmatrix} \mathbf{u}(\mathbf{q}),$$

where $\mathbf{u}(\mathbf{q})$ is the vector field given by (19). In this paper $d = 0.1\text{m}$.

Our experimental platform is composed by an iRobot's Create® mobile robot localized by a visual system composed by three calibrated overhead cameras that track the robot's position and orientation using the ArtoolKitPlus Tracking Library at 20Hz. The robot is equipped with a Linux based laptop and all the code was written in C++ based on the Player robot programming framework [28]. Figure 10 shows a picture of the robot.

Figure 11 shows the robot path as seen by the visual system when it is following a vector field determined by static functions of the form:

$$\begin{aligned} \alpha_1(x, y, z) &= ax^4 - bx^2y^2 + cy^4 - 1, \\ \alpha_2(x, y, z) &= z, \end{aligned}$$

where a , b , and c were chosen to fit the lab workspace. Not only for this experiment, but for all the experiments presented

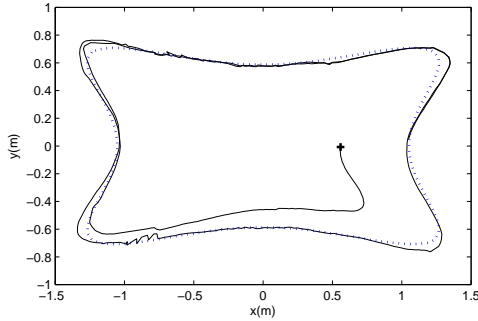


Fig. 11. Robot path (solid line) and desired static curve (dotted line) given a tie-like curve.

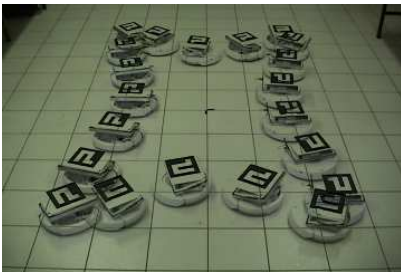


Fig. 12. Sequence of snapshots from the experiment in Figure 11.

in this section, the potential function was defined to be:

$$V = \sqrt{\alpha_1^2 + \alpha_2^2}.$$

Notice in Figure 11 that some small localization noise do not prevent the robot to circulate the desired curve shown as a dotted line. The difference between the robot path and the target curve can be explained by the distance d used in the feedback linearization controller. Figure 12 shows a sequence of snapshots of the experiments overlaid on the same picture to illustrate the actual robot behavior in the lab workspace.

In the second experiment presented here, the Create robot was subjected to a vector field that attracts it to the intersection of a paraboloid and a plane parallel to the z axis. As shown in Figure 13 the robot was initially positioned relatively far from the target curve.

In a third test with the Create robot, it is possible to see that the vector field in Equation (19) can be used to track a time-varying curve. The time-varying curve in this case is a circle with constant radius and center moving in the x direction with 0.01m/s. The circle is obtained by intersecting a moving cylinder with the $z = 0$ plane. Notice that this curve can be tracked by the robot since its maximum speed is 0.5m/s. Figure 14 shows the robot path for this experiment while Figure 15 shows the evolution of the potential function in time. Notice that V decreases very fast and, except for localization and actuation errors, remains very close to zero.

Our last experiment aims to show that the robot is able to track curves specified by a finite number of samples of the space. Figure 16 shows the robot path and the desired

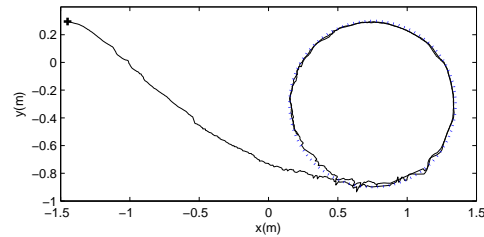


Fig. 13. Robot path (solid line) and desired static curve (dotted line) given by a static circle.

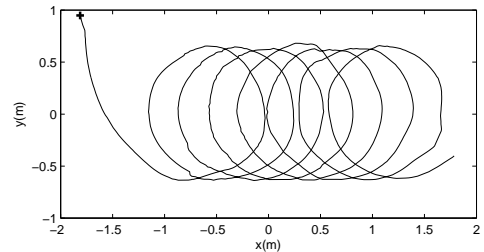


Fig. 14. Robot path when it is following a time-varying vector field determined by a moving circle.

curve. For this figure, the 39 samples in Figure 7 were used to generate α_1 while α_2 was given by a horizontal plane.

V. CONCLUSION

This paper presented a new methodology for robot navigation along a closed, generic-shaped curve using a continuous vector field. The field is given by the sum of three terms: (i) the gradient of a potential function; (ii) a circulation term; and (iii) a correction term (in the case of time-varying curves). These terms are computed by using a set of implicit functions that intersect at the target curve. We present proofs of convergence for point, single and double integration robots (see Appendix A for second-order dynamics). Our main contributions are related to the implicit function formulation of the problem and the inclusion of the correction term for time-varying curves. The target curve may be time-varying and may also be defined in n -dimensional configuration spaces.

The main limitation of the proposed approach is the fact that it is not fully automated. This is due to the need of ensuring that the set $\mathcal{U}(t)$ is repulsive and $\mathcal{T}(t) \cap \mathcal{U}(t) = \emptyset$ for all t . These are intrinsic properties of the functions α_i , and thus far, there is no automatic method that guarantee these hypotheses. Nevertheless, in Appendix B we show that if these functions are real analytic, then $\mathcal{U}(t)$ is a set of measure zero. Additionally, it is necessary to find the functions α_i . For a class of sample points of the desired curve, which satisfies Assumption 1, the paper does present a practical methodology to determine a set of functions. This technique relies on the interpolation of the samples using radial basis functions.

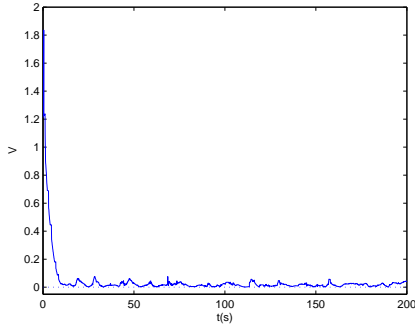
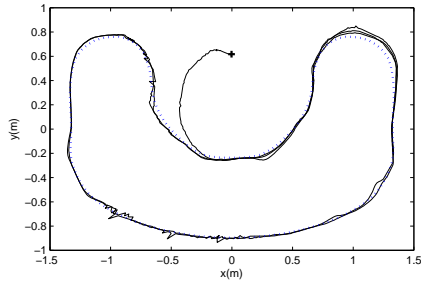

 Fig. 15. Function V for the path in Figure 14.


Fig. 16. Robot path (solid line) and desired static curve (dotted line) for a curve specified by 39 samples of the desired curve.

Another limitation is that the method is only guaranteed to control single and double integration dynamic systems. In Section IV-B we show a practical method to control a specific class of nonholonomic robots but, for other systems, further investigation is needed.

It is interesting to note that some approaches for pattern tracking and pattern generation previously found in the literature can be defined by the framework presented in this paper. If, for example, we define $V = \alpha_1^2/2$, $\alpha_1 = (x_1^2 + x_2^2 - r_d^2)/2$, and:

$$G = \frac{\beta v_0}{\sqrt{x_1^2 + x_2^2(x_1^2 + x_2^2 + r_d^2)}},$$

$$H = -\frac{2r_d\beta v_0}{(x_1^2 + x_2^2 + r_d^2)},$$

where β is a positive scaling factor, we have exactly the 2D vector field shown in [16] which allows for convergence and circulation of a loiter circle of radius r_d . Similar conclusions for the vector fields defined in [15] may be obtained. Also, the proposed methodology may be complementary to some other published works. In [19], for instance, the authors propose vector fields for n -dimensional spaces that guides UAVs to track the zero-level set of a potential function. The method proposed in the present paper could be used to compute such potential functions.

We are currently extending methodologies that control swarms of ground robots based on vector fields, such as [9] and [15], to 3-dimensional workspaces using the methodology proposed in this paper. This extension will be used to control

swarms of rotorcrafts. Results of a single rotorcraft tracking curves in 3D are presented in [26]. We also see several applications of the technique to guide teams of unmanned airplanes in tasks such as the ones proposed in [19].

Our future work also includes the extension of the methodology to consider obstacles. One possibility is to use a navigation function [1] or a harmonic function [3] as one of the intersecting functions that determines the target curve. Another extension, necessary to apply the methodology to guide fixed wing aerial robots, for example, is to develop techniques that impose constraints such as bounded curvature and torsion to the target curves.

APPENDIX A SECOND-ORDER DYNAMICS

The adaptation of vector field based approaches to control second-order mechanical systems is also of great interest. In [29], for example, such systems are addressed by adding a nonlinear dampening term to a harmonic potential field. In this appendix we present a result regarding second-order dynamic systems modeled by:

$$\ddot{\mathbf{q}} = \mathbf{u},$$

where \mathbf{u} is the system control input.

Consider the velocity vector field:

$$\mathbf{c} = -G\nabla_{\mathbf{q}}V + H \wedge_{i=1}^{n-1} \nabla_{\mathbf{q}}\alpha_i - M^{-1}\mathbf{a},$$

Consider also the definition:

Definition 13 Let $\kappa : \mathbb{R}^m \mapsto \mathbb{R}^m$ be a continuous function such that the system $\dot{\mathbf{p}} = \kappa(\mathbf{p})$ is globally asymptotically stable to the origin $\mathbf{p} = \mathbf{0}$.

We propose the following control law:

$$\mathbf{u} = J_{\mathbf{q}}(\mathbf{c})\dot{\mathbf{q}} + \frac{\partial \mathbf{c}}{\partial t} + \kappa(\dot{\mathbf{q}} - \mathbf{c}), \quad (26)$$

where $J_{\mathbf{q}}(\mathbf{c})$ is the Jacobian matrix relative to the variables x_1, x_2, \dots, x_n and κ is a function according to the Definition 13 with $m = n$.

If the hypotheses on Theorem 1 hold, then the system converges asymptotically to the set $\mathcal{T}(t)$ for any initial condition $\mathbf{q}_0, \dot{\mathbf{q}}_0$, provided that \mathbf{q}_0 is not in $\mathcal{U}(0)$. In addition, it is always possible to ensure circulation along it with a given fixed direction. Moreover, $\ddot{\mathbf{q}}$ is continuous.

To show this result, let $\mathbf{d} = \dot{\mathbf{q}} - \mathbf{c}$. Then, Equation (26) can be rewritten as:

$$\dot{\mathbf{d}} = \kappa(\mathbf{d}).$$

Therefore, for an initial condition \mathbf{d}_0 :

$$\mathbf{d}(t) = \zeta(t, \mathbf{d}_0).$$

According to the definition of \mathbf{d}

$$\dot{\mathbf{q}} = \mathbf{c} + \zeta(t, \mathbf{d}_0).$$

Since, by hypothesis, the origin is globally asymptotically stable for the system $\dot{\mathbf{d}} = \kappa(\mathbf{d})$, as t goes to ∞ the term ζ vanishes. According to Theorem 1, convergence and circulation along $\mathcal{T}(t)$ are then guaranteed.

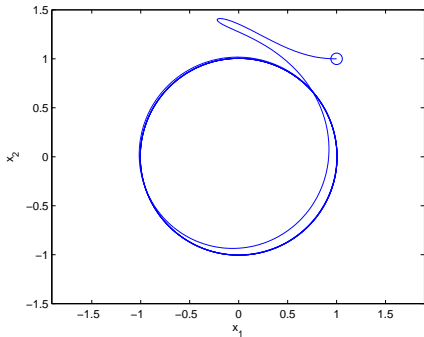


Fig. 17. Simulation of a second-order system traversing a circumference in \mathbb{R}^2 with initial configuration $\mathbf{q}_0 = [1 \ 1]^T$ and initial velocity $\dot{\mathbf{q}}_0 = [-3 \ 0]$.

A simulation is presented in Figure 17 with $n = 2$, initial position $\mathbf{q}_0 = [1 \ 1]^T$, and initial velocity $\dot{\mathbf{q}}_0 = [-3 \ 0]$. This initial velocity is not aligned to the vector field which implies a sinuous path towards the target curve.

APPENDIX B MEASURE OF THE REPULSIVE SET

If the functions α_i are real analytic almost everywhere $\forall t \geq 0$, then the set $\mathcal{U}(t)$ has zero measure in \mathbb{R}^n . As discussed in this text, $\mathcal{U}(t)$ can be seen as the zero set of the function $\mathbf{f} = \bigwedge_{i=1}^{n-1} \nabla \alpha_i : \mathbb{R}^n \mapsto \mathbb{R}^n$. We will show that if \mathbf{f} is real analytic almost everywhere and has a non-zero measure set such that $\mathbf{f} = \mathbf{0}$, then \mathbf{f} is zero almost everywhere (thus the functions α_i are not applicable for the algorithm, since circulation is impossible).

If the functions α_i are real analytic almost everywhere, it is clear that so is \mathbf{f} , since the product and sum of real analytic functions are also real analytic. Then, we use the following lemma:

Lemma 1 *Let $\mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^n$ be a real analytic function in a domain \mathcal{D} . Suppose that there exists a domain $\mathcal{N} \subseteq \mathcal{D}$ with non-zero measure such that $\mathbf{f} = \mathbf{0}$. Then, for all $\mathbf{x} \in \mathcal{N}$, all partial derivatives of all orders of \mathbf{f} are null.*

Proof: Let \mathbf{x} be a point in \mathcal{N} . Since \mathcal{N} has non-zero measure, is always possible to find a ball B centered in \mathbf{x} with a non-null radius $\epsilon(\mathbf{x})$ such that for all points in B we have $\mathbf{f} = \mathbf{0}$. Let

$$D_r(\mathbf{f}, \hat{\mathbf{x}}) = \frac{1}{r}(\mathbf{f}(\mathbf{x} + \hat{\mathbf{x}}r) - \mathbf{f}(\mathbf{x})),$$

in which r is a real number and $\hat{\mathbf{x}}$ is a unit vector. It is clear then, that $\forall \hat{\mathbf{x}}, \forall r \leq \epsilon$ we have $\mathbf{x} + \hat{\mathbf{x}}r \in B$ and then $\mathbf{f}(\mathbf{x} + \hat{\mathbf{x}}r) = \mathbf{f}(\mathbf{x}) = \mathbf{0}$, and so $D_r = \mathbf{0}$. In special

$$\lim_{r \rightarrow 0} D_r(\mathbf{f}, \hat{\mathbf{x}}) = J(\mathbf{f})\hat{\mathbf{x}} = \mathbf{0},$$

where $J(\mathbf{f})$ is the Jacobian of \mathbf{f} . Since $\hat{\mathbf{x}}$ is an arbitrary unit vector, all the columns of $J(\mathbf{f})$, $\partial \mathbf{f} / \partial x_i$, must vanish in the domain \mathcal{N} . Since \mathbf{f} is real analytic, so is $\partial \mathbf{f} / \partial x_i$. Then, the lemma hypothesis also hold and we can apply it successively to conclude that the derivatives of all orders vanish in \mathcal{N} . ■

The following Theorem can now be proved.

Theorem 2 *Let $\mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^n$ be a real analytic function in a domain \mathcal{D} . Suppose that there exists a domain $\mathcal{N} \subseteq \mathcal{D}$ with non-zero measure such that $\mathbf{f} = \mathbf{0}$. Then $\mathbf{f} = \mathbf{0}$ in \mathcal{D} .*

Proof: Lemma 1 indicates that there is a $\mathbf{x} \in \mathcal{N} \subseteq \mathcal{D}$ where all partial derivatives of \mathbf{f} vanish. A direct adaptation of the Theorem 3.13 of [30] to real analytic functions state that this fact implies that $\mathbf{f} = \mathbf{0}$ in \mathcal{D} . ■

Therefore, if \mathbf{f} is non null almost everywhere then at least one hypothesis of the theorem must fail: since it is real analytic almost everywhere by hypothesis, then its impossible to exist a set of non-zero measure such that $\mathbf{f} = \mathbf{0}$.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support of Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) – Brazil.

REFERENCES

- [1] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Automat.*, vol. 8, no. 5, pp. 501–518, 1992.
- [2] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.
- [3] C. I. Connolly, J. B. Burns, and R. Weiss, "Path planning using Laplace's equation," in *Proc. IEEE Int. Conf. Robotics Automation*, 1990, pp. 2102–2106.
- [4] L. C. A. Pimenta, G. A. S. Pereira, and R. C. Mesquita, "Fully continuous vector fields for mobile robot navigation on sequences of discrete triangular regions," in *Proc. IEEE Int. Conf. Robotics Automation*, 2007, pp. 1992–1997.
- [5] L. Zhang, S. M. LaValle, and D. Manocha, "Global vector field computation for feedback motion planning," in *Proc. IEEE Int. Conf. Robotics Automation*, 2009, pp. 477–482.
- [6] A. A. Masoud, "A harmonic potential field approach with a probabilistic space descriptor for planning in non-divisible environments," in *Proc. IEEE Int. Conf. Robotics Automation*, 2009, pp. 3774–3779.
- [7] G. A. S. Pereira, L. C. A. Pimenta, A. R. Fonseca, L. Q. Correa, R. C. Mesquita, L. Chaimowicz, D. S. C. Almeida, and M. F. M. Campos, "Robot navigation in multi-terrain outdoor environments," *Int. J. Robot. Res.*, vol. 28, pp. 685–700, 2009.
- [8] W. J. Pisano, D. A. Lawrence, and K. Mohseni, "Concentration gradient and information energy for decentralized uav control," in *AIAA Guidance, Navigation, and Control Conf.*, 2006.
- [9] L. Chaimowicz, N. Michael, and V. Kumar, "Controlling swarms of robots using interpolated implicit functions," in *Proc. IEEE Int. Conf. Robotics Automation*, 2005, pp. 2498–2503.
- [10] M. A. Hsieh and V. Kumar, "Pattern generation with multiple robots," in *Proc. IEEE Int. Conf. Robotics Automation*, 2006, pp. 2442–2447.
- [11] L. C. A. Pimenta, M. L. Mendes, R. C. Mesquita, and G. A. S. Pereira, "Fluids in electrostatic fields: An analogy for multi-robot control," *IEEE Trans. Magn.*, vol. 43, pp. 1765–1768, 2007.
- [12] P. Y. Li and R. Horowitz, "Passive velocity field control of mechanical manipulators," *IEEE Trans. Robot. Automat.*, vol. 15, no. 4, pp. 751–763, 1999.
- [13] J. Moreno-Valenzuela, "On passive velocity field control of robot arms," *J. Frankl. Inst.*, vol. 344, no. 8, pp. 1021–1038, 2007.
- [14] M. Quigley, M. A. Goodrich, S. Griffiths, A. Eldredge, and R. Beard, "Target acquisition, localization, and surveillance using a fixed-wing mini-UAV and gimbaled camera," in *Proc. IEEE Int. Conf. Robotics Automation*, 2005, pp. 2600–2605.
- [15] M. A. Hsieh, S. Loizou, and V. Kumar, "Stabilization of multiple robots on stable orbits via local sensing," in *Proc. IEEE Int. Conf. Robotics Automation*, 2007, pp. 2312–2317.
- [16] E. W. Frew, D. A. Lawrence, C. Dixon, J. Elston, and W. J. Pisano, "Lyapunov guidance vector fields for unmanned aircraft applications," in *Proc. American Control Conf.*, 2007, pp. 371–376.

- [17] F. Zhang and N. E. Leonard, "Coordinated patterns of unit speed particles on a closed curve," *Syst. Control Lett.*, vol. 56, no. 6, pp. 397–407, 2007.
- [18] N. Ceccarelli, M. D. Marco, and A. Giannitrapani, "Collective circular motion of multi-vehicle systems," *Automatica*, vol. 44, pp. 3025–3035, 2008.
- [19] D. A. Lawrence, E. W. Frew, and W. J. Pisano, "Lyapunov vector fields for autonomous unmanned aircraft flight control," *J. Guid. Control Dyn.*, vol. 31, no. 5, pp. 1220–1229, 2009.
- [20] S. Hernandez and D. A. Paley, "Stabilization of collective motion in a time-invariant flowfield on a rotating sphere," in *Proc. American Control Conf.*, 2009, pp. 623–628.
- [21] P. Iscold, G. A. S. Pereira, and L. A. B. Torres, "The development of a hand-launched small UAV for ground reconnaissance," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, pp. 335–348, 2010.
- [22] V. M. Gonçalves, L. C. A. Pimenta, C. A. Maia, and G. A. S. Pereira, "Artificial vector fields for robot convergence and circulation of time-varying curves in n-dimensional spaces," in *Proc. American Control Conf.*, 2009, pp. 2012–2017.
- [23] G. Turk and J. F. Brien, "Modelling with implicit surfaces that interpolate," *ACM Trans. Graph.*, pp. 855–873, 2002.
- [24] M. P. do Carmo, *Differential Forms and Applications*. Springer-Verlag, 1994.
- [25] J.-J. Slotine and W. Li, *Applied Nonlinear Control*. Prentice Hall, 1991.
- [26] V. M. Gonçalves, L. C. A. Pimenta, C. A. Maia, G. A. S. Pereira, B. C. O. Dutra, N. Michael, J. Fink, and V. Kumar, "Circulation of curves using vector fields: actual robot experiments in 2d and 3d workspaces," in *Proc. IEEE Int. Conf. Robotics Automation*, 2010, pp. 1136–1141.
- [27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.
- [28] B. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *Proc. Int. Conf. Advanced Robotics*, 2003, pp. 317–323.
- [29] A. A. Masoud, "Agile, steady response of inertial, constrained holonomic robots using nonlinear, anisotropic dampening forces," in *Proc. IEEE Conf. Decision and Control*, 2006, pp. 6167–6172.
- [30] B. Fuks, *Theory of Analytic Functions of Several Complex Variables*. American Mathematical Society, 1963.



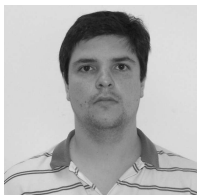
Carlos A. Maia received the B.S. and the M. S. degrees in Electrical Engineering from Universidade Federal de Minas Gerais (UFMG), Brazil. His Ph.D. was obtained at the Université d'Angers (France) and at the Universidade Estadual de Campinas (UNICAMP) (Brazil). Since 1997 he is an assistant professor of electrical engineering at the UFMG. His current research includes modeling, analysis and control of time and event-driven dynamical systems.



Bruno C. O. Dutra was born in Belo Horizonte, Minas Gerais, Brazil. At age 21 he is an undergraduate student of Electrical Engineering at Universidade Federal de Minas Gerais (UFMG), where, since February 2009, he conducts researches in the areas of computer-aided simulations and cooperative robotics at the Computer Systems and Robotics (CORO) Laboratory under the guidance of Professor Dr. Guilherme A. S. Pereira.



Vinícius M. Gonçalves was born in Contagem, Minas Gerais, Brazil in 1987. He is currently in process of obtaining his B.S. degree in Control and Automation Engineering from the Universidade Federal de Minas Gerais (UFMG). In 2009 he was a finalist for the Best Student Paper Award at the American Control Conference. His current research interests include robot navigation and discrete event systems.



Luciano C. A. Pimenta received the B.S., M.Sc., and Ph.D. degrees in electrical engineering from the Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brazil, in 2003, 2005, and 2009, respectively. From April 2007 to June 2008, he was a visiting Ph.D. student at the General Robotics, Automation, Sensing and Perception (GRASP) Laboratory at the University of Pennsylvania, Philadelphia, USA. He is currently an Assistant Professor with the Department of Electronic Engineering at UFMG. His research interests include robotics, multiagent

coordination, and control theory.



Guilherme A. S. Pereira received the B.S. and M.S. degrees in electrical engineering and the Ph.D. degree in computer science from the Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil, in 1998, 2000, and 2003, respectively. Dr. Pereira received the Gold Medal Award from the Engineering School of UFMG for garnering first place among the electrical engineering students in 1998. He was, from November 2000 to May 2003, a Visiting Scientist at the General Robotics, Automation, Sensing and Perception (GRASP) Laboratory, University of Pennsylvania, Philadelphia. Since July 2004, he is an Assistant Professor of the Electrical Engineering Department at the Federal University of Minas Gerais (DEE/UFMG), where he is the director of the Computer Systems and Robotics (CORO) Laboratory, one of the laboratories that compose the Group for Research and Development of Autonomous Vehicles (PDVA) at UFMG. His research interests include cooperative robotics, robot navigation, autonomous vehicles development, computer vision, and distributed sensing. Dr. Pereira is a Member of Sociedade Brasileira de Automática.